

**In the Claims:**

Please amend the claims as follows. The following listing of claims replaces all previous listings.

1. (Previously Presented) A method comprising:

responsive to beginning a rich internet application (RIA) and deferring instantiation of one or more interface elements in said RIA, generating a descriptor tree having a plurality of descriptor nodes, wherein each of said plurality of descriptor nodes describes a plurality of interface elements of said RIA instantiated and visible at said beginning of said RIA;

creating one or more hidden descriptor nodes in said descriptor tree describing said one or more interface elements not instantiated or visible at said beginning, wherein said hidden descriptor nodes are created responsive to a user navigating to said one or more interface elements and wherein, further responsive to said user navigating, instantiating said interface elements not instantiated at said beginning of said RIA; and

rendering said plurality of interface elements instantiated at said beginning of said RIA and said one or more interface elements not instantiated at said beginning of said RIA using corresponding ones of:

said plurality of descriptor nodes; and

said one or more hidden descriptor nodes.

2. (Previously Presented) The method of claim 1 further comprising:

converting said plurality of descriptor nodes into a plurality of detail objects;

converting said one or more hidden descriptor nodes into one or more detail objects not instantiated at said beginning of said RIA, wherein said plurality of interface elements instantiated at said beginning of said RIA and said one or more interface elements not

instantiated at said beginning of said RIA are rendered directly using said plurality of detail objects and said one or more detail objects not instantiated at said beginning of said RIA.

3. (Original) The method of claim 1 wherein each one of said plurality of descriptor nodes and said one or more hidden descriptor nodes contains a software method for generating each its child nodes.

4. (Original) The method of claim 1 further comprising:  
downloading executable code representing said RIA to a computer of said user responsive to said beginning of said RIA, wherein said generating and said creating use said executable code.

5. (Previously Presented) The method of claim 2 further comprising:  
storing as a plurality of stored nodes each of:  
said plurality of descriptor nodes;  
said one or more hidden descriptor nodes;  
said plurality of detail objects; and  
said one or more detail objects not instantiated at said beginning of said RIA; and  
re-rendering each of said plurality of interface elements instantiated at said beginning of said RIA and said one or more interface elements, not instantiated at said beginning of said RIA, from said plurality of stored nodes.

6. (Previously Presented) The method of claim 1 wherein said one or more hidden descriptor nodes created has a navigational relationship with a particular one of said one or more hidden interface elements, not instantiated at said beginning of said RIA, to which said user navigates.

7. (Original) The method of claim 6 wherein said navigational relationship

comprises one or more of:

a direct link;

an ordinal relationship;

a statistical relationship; and a positional relationship.

8. (Original) The method of claim 1 further comprising:

creating select ones of said one or more hidden descriptor nodes in said descriptor tree responsive to beginning said RIA.

9. (Previously Presented) A method comprising:

creating a root application node of a descriptor tree, responsive to a user initiating a rich internet application (RIA) defined using procedural code and declarative code;

generating a plurality of descriptor nodes for said descriptor tree, wherein each of said plurality describes an interface element currently instantiated and visible to said user on a currently visible pane of said RIA;

responsive to said user navigating to a subsequent pane of said RIA, constructing a plurality of stacked descriptor nodes for said descriptor tree, wherein each of said plurality of stacked descriptor nodes describes said interface element not instantiated and invisible to said user on said currently visible pane of said RIA and associated with said subsequent pane; and

creating a detail object from each one of:

said plurality of descriptor nodes; and

said plurality of stacked descriptor nodes; and

rendering said interface element using a corresponding detail object.

10. (Previously Presented) The method of claim 9 wherein said generating comprises:

generating one of said plurality of descriptor nodes for a container of said interface element not instantiated and invisible to said user on said currently visible pane of said RIA.

11. (Original) The method of claim 9 wherein said association between said subsequent pane and said plurality of stacked descriptor nodes comprises one of:

a direct link;

an ordinal relationship;

a statistical relationship; and a positional relationship.

12. (Currently Amended) A computer program product having a computer readable medium with computer program logic recorded thereon, said computer program product comprising:

code for initializing of a rich internet application (RIA), wherein initializing comprises instantiating at least one visible element in said RIA and deferring instantiation of unseen ones of a plurality of stacked elements in said RIA;

~~responsive to starting a rich internet application (RIA) and deferring instantiation of unseen ones of a plurality of stacked elements in said RIA,~~ code for generating a descriptor tree having a plurality of descriptor nodes, wherein each of said plurality of descriptor nodes describes a plurality of instantiated and visible interface elements of said RIA;

code for creating one or more stacked descriptor nodes in said descriptor tree describing said unseen ones of said plurality of stacked interface elements responsive to a user navigating to said unseen ones, wherein said unseen ones are not instantiated at said starting; and

code for rendering said plurality of visible interface elements and said unseen ones using corresponding ones of:

said plurality of descriptor nodes; and

said one or more stacked descriptor nodes.

13. (Previously Presented) The computer program product of claim 12 further comprising:

code for converting said plurality of descriptor nodes into a plurality of detail objects;

code for converting said one or more stacked descriptor nodes into one or more stacked detail objects, wherein said plurality of instantiated and visible interface elements and said unseen ones are rendered directly using said plurality of detail objects and said one or more stacked detail objects.

14. (Original) The computer program product of claim 12 wherein each one of said plurality of descriptor nodes and said one or more stacked descriptor nodes contains a software method for generating each its child nodes.

15. (Currently Amended) The computer program product of claim 12 further comprising:

code for downloading bytecode representing said RIA to a computer of said user responsive to said ~~starting~~ initializing of said RIA, wherein said code for generating and said code for creating use said bytecode.

16. (Previously Presented) The computer program product of claim 13 further comprising:

code for storing as a plurality of stored nodes each of:

said plurality of descriptor nodes;

said one or more stacked descriptor nodes; said plurality of detail objects; and

said one or more stacked detail objects; and

code for re-rendering each of said plurality of instantiated and visible interface elements and said one or more stacked interface elements from said plurality of stored nodes.

17. (Original) The computer program product of claim 12 wherein said one or more stacked descriptor nodes created has a navigational relationship with a particular one of said one or more stacked interface elements to which said user navigates.

18. (Original) The computer program product of claim 17 wherein said navigational relationship comprises one or more of:

a direct link;

an ordinal relationship;

a statistical relationship; and a positional relationship.

19. (Original) The computer program product of claim 12 further comprising:

code for code for creating select ones of said one or more stacked descriptor nodes in said descriptor tree responsive to starting said RIA.

20 – 27. (Cancelled).

28. (New) A computer program product having a computer readable medium with computer program code recorded thereon, said computer program product comprising:

program code for accessing executable code of a rich internet application, the executable code comprising code for instantiating a plurality of objects, each object for rendering a corresponding interface element of the rich internet application;

program code for identifying a subset of the plurality of objects in the executable code, the subset comprising fewer than all of the plurality of objects;

program code for instantiating the objects in the subset;

program code for rendering an initial view of the application using the instantiated objects;

program code for instantiating at least one other object of the plurality of objects in response to user interaction with an interface element of the initial view; and

program code for rendering another view of the application using the instantiated at least one other object.

29. (New) The computer program product set forth in claim 28, further comprising program code for creating a descriptor tree comprising a plurality of descriptor nodes, each node identifying an object or container of the application,

wherein identifying the subset comprises (i) using the descriptor tree to identify a node whose object is for rendering an interface element that is not visible in the initial view and (ii) excluding the object from the subset of objects.

30. (New) The computer program product set forth in claim 29, wherein using the descriptor tree to identify a node whose object is for rendering an interface element that is not visible in the initial view comprises:

determining if the node identifies itself as corresponding to a stacked navigation object.

31. (New) The computer program product set forth in claim 29, wherein creating a descriptor tree comprising identifying a node whose object is for rendering an interface element that is not visible in the initial view as a hidden node; and wherein objects associated with hidden nodes are excluded from the subset.